Remarks

The present amendment responds to the Official Action dated March 29,2004. A petition for a one month extension of time to respond and authorization to charge Deposit Account No. 50-1058 the large entity extension fee of $ 110 accompany this amendment. The Official Action objected to claims 1 and 6 as being unclear. Claim 1 was rejected under 35 U.S.C. §112 on the grounds that certain terms lacked antecedent basis. Claims 1, 7, 8, and 11 were rejected under 35 U.S.C. §102(b) based on Lee et al. U.S. Patent No. 4,763,242 ("Lee").

Dependent claims 3 and 4 were rejected under 35 U.S.C. §103(a) as being unpatentable over Lee in view of Dowling U. S. Patent No. 6,128,728 ("Dowling '728"). Dependent claim 5 was rejected under 35 U.S.C. §103(a) as being unpatentable over Lee in view of Dowling U.S. Patent No. 6,170,051 ("Dowling '051").

Claim 6 was objected to as being dependent upon a rejected base claim but was indicated to be allowable if rewritten in independent form.

Claims 8-11 were rejected under 35 U.S.C. §103(a) as being unpatentable over Dahl et al. U.S. Patent No. 5,710,938 ("Dahl") in view of Lee. Dependent claims 12 and 13 were rejected under 35 U.S.C. §103(a) as being unpatentable over Lee, in further view of Mirsky et al. U.S. Patent No. 5,915,123 ("Mirsky"). These grounds of rejection are addressed below.

Claims 1, 6, 8, 9, 10, and 11 have been amended to be more clear and distinct. A proposed amendment to claim 1 was discussed in a telephone interview with the Examiner summarized below. In the interview, the Examiner agreed the proposed amendment defined over the relied upon art. Claim 1 has been amended according to the agreed upon proposed

amendment discussed in the telephone interview placing it in order for allowance. Claim 6 has been rewritten in independent form placing it in order for allowance. Independent claims 8 and 11 have been amended to include similar limitations regarding the CSB and the SP/PE selection bit as presently amended claim 1. Claims 1 and 3-13 are presently pending with claims 1 and 6 standing in order for allowance.

Objections to claims 1 and 6

Claim 1 was objected to because the phrase "the SP by one PE array processor environment" was unclear. Claim 1 has been amended to replace the objected to phrase with the phrase "the merged SP and PE processor environment" to reflect that the SP and PE are merged and "...can be viewed as a single processor containing two register contexts that share a common set of execution units." See the Specification, page 3, lines 1-10.

Claim 6 was objected to because the phrase "... to determine whether which register files the SP's register files or the PE's register files are to be accessed..." is not grammatically correct. Claim 6 has been amended to replace the objected to phrase with the phrase " ...to determine whether the SP's register files or the PE's register files are to be accessed..."

Antecedent Basis Rejection of Claim 1

Claim 1 has been amended to recite "an SP register file" in claim 1 line 6, and "a PE register file" in claim 1 line 7 to provide an antecedent for the SP register file and the PE register file, respectively.

Telephone Interview Summary

The Examiner is thanked for the courtesy of a telephone interview conducted on June 29, 2004. Dr. Pechanek, a co-inventor, participated with Mr. Agusta in this call. Prior to the telephone interview, a proposed amendment to claim 1, attached as Exhibit A, was faxed to the Examiner prior to the interview for review. In the telephone call, the proposed amendment to claim 1 was discussed. One aspect of the amendment distinctly claims an SP/PE selection bit in an instruction register, that can change its value with each fetched instruction, and a CSB in a state register, that persists between fetched instructions. The combination of the SP/PE selection bit and the CSB selects a context of a first software task using a first array configuration or a context of a second software task using a second array configuration. Dr. Pechanek further explained his invention to the Examiner in order to clear up some confusion in the understanding of the configuration of the processing elements.

The Examiner agreed that the proposed amendment appears to define over the relied upon art. The Examiner pointed out the 35 U.S.C. §112 rejection regarding the proposed claim 1 amendment which we agreed to correct. The Examiner further pointed out the wording "a fetch controller for fetching a single instruction stream from a plurality of instructions" in the proposed amendment to claim 1 was potentially unclear, but accepted the suggestion of changing the phrase to "a fetch controller for fetching a plurality of instructions".

The Art Rejections

As addressed in greater detail below, Lee or various combinations of Lee with Dowling '728, Dowling '051, Dahl, or Mirsky do not support the Official Action's reading of the relied upon art and the rejections based thereupon should be reconsidered and withdrawn. Further, the Applicant does not acquiesce in the analysis of the relied upon art made by the Official Action and respectfully traverses the Official Action's analysis underlying its rejections.

Lee describes a system which includes a main processor and an assist. An assist as determined by Lee is hardware that extends the main processor's capabilities by providing support for additional extension instructions which are not part of the main processor's basic instruction set. The Lee assist may be in the form of a coprocessor or a special function unit. When an instruction is fetched from memory, a field in the instruction is decoded to determine whether the instruction is a basic instruction or an extension instruction. If the instruction is a basic instruction, it is executed by the main processor. If the instruction is an extension instruction, the field is further decoded to determine which assist to route the instruction to for execution. Lee makes no determination of which register file to use during the execution of the instruction. See Lee col. 4 lines 48-68 and Fig. 1, components 129, 131, and 133 of instruction 127.

In contrast to Lee, the present invention provides techniques which result in efficient context switching between software tasks executing in an array processing environment. A register context defines which register file, such as a sequence processor (SP) register file or a

processing element (PE) register file, is used when executing an instruction. For example, one register context may include an SP utilizing data stored in a PE register when executing an SP instruction. In one aspect of the present invention, an SP is merged with a PE and execution units are shared by the SP and PE. Consequently, in the merged SP/PE, a single set of execution units are coupled with two independent register files, the SP register file and the PE register file. To make efficient use of the SP and PE resources and obtain multiple register contexts, the present invention provides two independent bits: an SP/PE bit carried in a fetched instruction to differentiate SP instructions from PE instructions and a context switch bit (CSB) which is stored in a processor state register. The CSB value persists between fetched instructions. The control of the SP/PE bit in conjunction with the CSB advantageously determines the desired register context.

In one exemplary operation in accordance with the present invention, a fetched instruction containing the SP/PE bit is received in an instruction register that provides control information for the execution of a fetched instruction. Each time an instruction is fetched, the SP/PE bit value in the instruction register may change depending on the fetched instruction. The CSB bit is modified by a specific instruction used specifically for the purposes of loading the processor state register. Consequently, the CSB bit value persists between fetched instructions for program execution. The CSB value, that persists between fetched instructions, in conjunction with the SP/PE bit which may change with each fetched instruction, allows the selection between the PE register file or the SP register file for use by a sequential instruction. The CSB value in conjunction with the SP/PE bit may further advantageously support reconfiguration of PE and SP

resources to address software that favors sequential instruction processing. In addition, this arrangement may advantageously allow execution units to be shared between the usage of the SP and PE register files. See amended claim 1, for example, which reads as follows:

> 1. Apparatus for providing efficient context switching between software tasks in a merged sequence processor (SP) and processor element (PE) processor environment, each software task comprising a plurality of instructions, the merged SP and PE processor environment configurable to be in a first array configuration or a second array configuration, the apparatus comprising:
>
> a first set of registers stored in an SP register file;
>
> a second set of registers stored in a PE register file;
>
> an execution unit that is shared to execute SP instructions and PE instructions;
>
> a fetch controller for fetching a plurality of instructions;
>
> an instruction register for receiving each fetched instruction, the instruction register providing control information for the execution of a fetched instruction, the instruction register having a sequence processor/processing element (SP/PE) selection bit set by a fetched instruction whereby the SP/PE selection bit value can change with each fetched instruction; and
>
> a processor state register having a context select bit (CSB), a specific instruction out of the plurality of instructions setting the CSB value, the CSB being independent of bits in the instruction register, the CSB value persisting between the fetched instructions, the CSB value in conjunction with the SP/PE selection bit value selecting a context of a first software task utilizing the first array configuration or a context of a second software task utilizing the second array configuration, the first array configuration including at least one register from the second set of registers to execute sequential instructions on the execution unit, the second array configuration including at least one register from the first set of registers to execute sequential instructions on the execution unit. (emphasis added)

See also amended claims 8 and 11 which include the limitations of the SP/PE bit and CSB as being independent register bits that change state by distinctly different techniques.

Lee does not teach and does not render obvious such techniques for sharing register files by means of a bit in a processor state register that persists between fetched instructions in

conjunction with a bit in an instruction register that may change with each fetched instruction as presently claimed. Further, Lee does not allow an execution unit to share register files. In fact, Lee teaches away from sharing of register files by always using the register file within the main processor or an assist, whichever executes the instruction. Thus, Lee does not provide sharing of register files between basic instructions and extension instructions as presently claimed. Lee merely allocates basic instructions to be executed on the main processor and extension instructions to be executed on the assist.

The Official Action asserts that the assist bit field described at col. 4, lines 66-68 of Lee constitutes "a context select bit (CSB)" that is stored "in a processor state register". This analysis is incorrect as the claimed CSB is distinctly different than Lee's assist bit field. Lee's use of the assist bit field is to indicate which hardware assist should be used for executing the instruction. It is not a context select bit stored in a processor state register that is independent of an instruction register, as presently claimed.

The Official Action in the Response to Arguments section further asserts that "... nothing is mentioned about sharing or, for instance, when a PE instruction is executing, the SP register file is used, and vice versa." Applicants respectfully disagree. However, claim 1, as presently amended, clearly recites sharing as follows: "... an execution unit that is shared to execute SP instructions and PE instructions;" and "...the first array configuration including at least one register from the second set of registers to execute sequential instructions on the execution unit, the second array configuration including at least one register from the first set of registers to execute sequential instructions on the execution unit."

The Official Action also further rejected previous arguments regarding the role of the SP/PE selection bit in the IR and the CSB in a processor state register. Claims 1, 8, and 11 have been amended to make clear and distinct the SP/PE selection bit, the CSB, and the use of "...the CSB value in conjunction with the SP/PE selection bit value selecting a context of a first software task utilizing the first array configuration or a context of a second software task utilizing the second array configuration."

Regarding claims 8-11, Dahl does not cure the failings of Lee as a reference. Dahl teaches an array system in which the array is partitioned into multiple sub-arrays which operate independently of each other. In Fig. 1 of Dahl, the array 11 is configured by a control processor 20. The control processor 20 does not have direct access to register files of any processor node in array 11 as it communicates with the processor nodes and network nodes via bit-serial control channels for the purpose of partitioning the array. See Dahl, col. 4, lines 3-6 and lines 11-16. Consequently, no processor node resources are shared with other processor nodes. As admitted in the Official Action, Dahl does not teach apparatus for providing efficient context switching between tasks. As addressed above, Lee also fails to teach context switching between tasks as presently claimed. Thus, the proposed combination of these two items does not teach and does not make obvious the present claims.

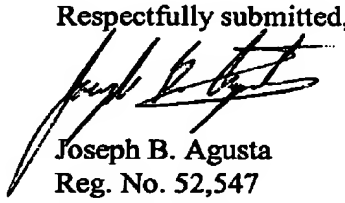
In summary, the relied upon art does not indicate a recognition of the problems addressed by the present invention. Further, the relied upon art does not teach and does not suggest an apparatus which would solve the problems of context switching on array processors addressed by

the present invention in the manner solved by the present invention. The claims as presently amended are not taught, are not inherent, and are not obvious in light of the relied upon art.

Conclusion

All of the presently pending claims, as amended, appearing to define over the applied references, withdrawal of the present rejection and prompt allowance are requested.

Respectfully submitted,



Joseph B. Agusta
Reg. No. 52,547
Priest & Goldstein, PLLC
5015 Southpark Drive, Suite 230
Durham, NC 27713-7736
(919) 806-1600

Exhibit A

1. (currently amended): Apparatus for providing efficient context switching between software tasks in a merged sequence processor (SP) and processor element (PE) processor environment, each software task comprising a plurality of instructions, the merged SP and PE processor environment configurable to be in a first array configuration or a second array configuration, the apparatus comprising:

a first set of registers stored in the SP register file;

a second set of registers stored in the PE register file;

an execution unit that is shared to execute SP instructions and PE instructions;

a fetch controller for fetching a single instruction stream from a plurality of instructions,

an instruction register for receiving each fetched instruction, the instruction register providing control information for the execution of a fetched instruction, the instruction register having a sequence processor/processing element (SP/PE) selection bit set by a fetched instruction whereby the SP/PE selection bit value can change with each fetched instruction; and

a processor state register having a context select bit (CSB) , a specific instruction out of the plurality of instructions setting the CSB value, the CSB being independent of the instruction register, the CSB value persists between the fetched instructions, the CSB value in conjunction with the SP/PE selection bit value selecting a context of a first software task utilizing the first array configuration or a context of a second software task utilizing a second array configuration, the first array configuration including at least one register from the second set of registers to execute sequential instructions on the execution unit, the second array configuration including at least one register from the first set of registers to execute sequential instructions on the execution unit.